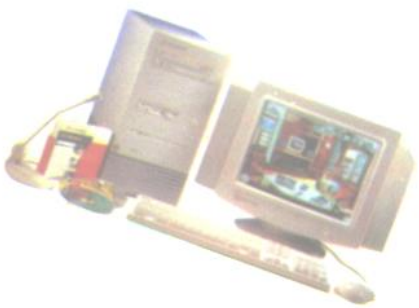
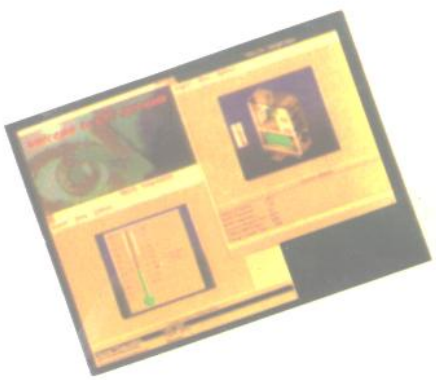
四川省计算机基础教育研究会推荐系列教材

PASCAL

语言程序设计

彭启琮　李玉柏　编著
李光琳　主审





电子科技大学出版社

PASCAL 语言程序设计
彭启琮 李玉柏 编著

四川省计算机基础教育研究会推荐系列教材

PASCAL 语言程序设计

丛书顾问 兰家隆 古天祥 王正智
黄迪明 梁光春 惠 林

丛书编委会 张慧云 陆 成 李光琳 向孟光
谢惠娟 蔡学望 吴家培 杨国才
黎 明 杨明广

彭岩琮 李玉柏 编 著
李光琳 主 审

电子科技大学出版社

内容提要

本书是为大专院校非计算机专业学生编写的计算机程序设计课程的教材，也是计算机等级考试中程序设计部分的参考教材。

本教材以国内外广泛使用的优秀程序设计教学语言 PASCAL 为基础，深入浅出地介绍结构化程序设计的基本思想和方法，力图使学生从一开始起就养成良好的程序设计的风格和习惯，为进一步的学习和实践打好基础。全书共分八章。充分考虑到初学者的知识结构和能力，全书非常注意讲述的深入浅出和循序渐进，并辅以适当的例题和习题，使学生学完本课程后，其程序设计的能力有较大的提高。

本书可以作为其他学校以及计算机培训班的程序设计课程教材，也可作为有关读者的自学教材。

四川省计算机基础教育研究会推荐系列教材

PASCAL 语言程序设计

彭启琮　李玉柏　编著

李光琳　主审

*

电子科技大学出版社出版

（成都建设北路二段四号）　邮编 610054

四川建筑印刷厂印刷

新华书店经销

*

开本 787×1092　1/16　印张 13.375 字数 325 千字

版次 1997 年 2 月第一版　印次 1997 年 2 月第一次印刷

印数 1—6000 册

ISBN 7—81043—652—X/TP・258

定价：14.50 元

序

随着科学技术的飞速发展，计算机已成为各行各业不可缺少的应用工具，计算机知识和应用能力已成为当代大学生培养质量高低的一个重要组成部分，是面向21世纪的高等教育培养高质量人才最突出需要加强的教学环节之一。目前在全国高校中普遍开展的计算机知识和应用能力等级考试制度正有效地推动这一目标的实现，四川省已成功地举行了五次这样的考试正充分地证明了这一点，而已成立三年多的四川省计算机基础教育研究会为保证我省计算机知识和应用能力等级考试制度和方法的不断完善提供了有力的保证。

到目前为止，有关计算机应用等级考试的图书不少，对推动我省计算机知识和应用能力的普及起了积极的作用，但由于他们中的大多数是个别作者独立写作，难以反映众多学生对教材的广泛需求。这一套计算机等级考试二级系列通用教材及配套复习资料和实验指导书一共十五种的主编和执笔者是来自省内许多大专院校，他们均是长期工作在该领域教学第一线，有丰富教学经验的优秀教师，他们中大多数是四川省计算机基础教育研究会成员单位的代表，他们的

参与为本套教材的质量和水平提供了可行的保证，也使本套教材有十分广泛的代表性和符合四川省等级考试大纲的要求。电子科技大学出版社为这套系列教材的成功编写作了大量的组织工作。

综上所述，本系列教材可作为我省高校非计算机专业计算机等级考试(二级)的教材或参考书。我相信，这套系列教材的出版将进一步推动我省高校计算机基础教育质量的提高，推动计算机应用的进一步普及，对我国全民族现代化素质的进一步提高有所裨益。

四川省计算机等级考试委员会副主任

兰家隆

1997年元月30日于成都

前　言

随着计算机科学与技术的发展，计算机的应用已渗透到科学技术、政治、经济、军事、文化、教育以及社会生活的方方面面。现代的大学生，无论现在学习什么专业，将来从事什么职业，都应该充分重视和建立“计算机意识”，要像学习语文、数学、外语一样，学习和掌握有关计算机的基础知识和基本技术。否则，就会成为信息时代的“文盲”。这一点已成为教育行政管理部门、教师、学生们的共识。“计算机文化”的概念正在逐步形成，并被人们广泛地接受。

尽管人们对非计算机专业学生的计算机教育应该深入到什么程度，尚有不尽相同的认识，但绝大多数人认为，学习并掌握一门以上的计算机语言，熟练地编写应用程序，并用以解决实际问题，是对理工科大学生进行计算机基础教育必要的基本要求之一。

接下来的问题就是，应该选择什么样的计算机语言，来作为程序设计的入门语言。

在计算机语言发展的不长的历史中，出现过多种语言，但真正得到广泛接受和使用的并不多。

作为第一个体现结构化程序设计思想的高级程序设计语言，PASCAL 语言设计的初衷，是建立一种有益于教学，且易于实现的程序设计语言。二十多年的历史已经证明，由于PASCAL 语言的语句简明、数据类型丰富、程序结构严谨，用它来学习程序设计，有助于培养良好的的程序设计的习惯和风格，因而被公认为培养程序设计人才的最佳教学语言。

本书就是为非计算机专业的学生编写的PASCAL 语言程序设计教材，也是四川省计算机分级考试的指定教材。

本书第一章至第四章由彭启琮编写，其余各章由李玉柏编写。与本教材配套的还有由惠林老师编写的同名实验教材，供指导上机用。

本教材由四川联合大学李光琳副教授主审。李教授提出了许多中肯的意见和建议。四川省计算机分级考试委员会和电子科技大学出版社对本书的出版给予了很大的帮助，在此一并表示感谢。

计算机科学和技术正在迅速地发展，对我们提出了越来越高的要求。由于编者的水平所限，书中的不当和错误之处在所难免，恳切地希望专家和读者批评指正。

编　者

1996 年9 月于电子科技大学

目　录

第一章　绪论

第二章　简单的PASCAL 程序

第三章 流程控制

第四章 函数与过程

第五章　枚举类型、子界类型、数组类型

第六章　算法初步与结构化程序设计方法

第七章　集合、记录和文件

第八章 指针与动态数据结构

第一章 绪 论

§1.1 为什么要学习计算机程序设计

1.1.1 计算机意识和计算机文化

人们目前广泛谈论的新的工业革命、信息时代等，都是以计算机技术和信息科学为基础和标志的。电子计算机在本世纪中叶的出现、发展，以及迅速地普及应用，给我们的科学技术、产业、社会的发展，乃至人们的生活方式和思维方式等都带来了极大的影响。

因此，有关计算机的知识已和语文、数学、物理、外语等一样，成为现代人知识结构中不可缺少的重要组成部分。正是基于科学技术和社会的发展对各种学科，不光是计算机学科，也不光是理工科，而是所有学科的学生在计算机知识和能力方面提出的越来越高的迫切要求，人们提出了所谓“计算机意识”和“计算机文化”的概念。

不会读书识字者是文盲。现在，一点也不具备计算机知识，不会使用计算机，就会成为“信息时代的文盲”。这一点已被越来越多的人所认同。无论学习什么专业的学生，无论他(她)将来希望从事什么样的职业，都强烈地意识到，必须学习、掌握一定的计算机知识和使用计算机的本领，才能从容地面对目前的学习和将来的职业需求，否则就会被淘汰出局。这就是计算机意识。

既然计算机在我们的社会中扮演着如此重要的角色；既然所有的人，无论从事什么职业，处于什么职务，都必须学习与掌握计算机知识和使用计算机的技能，那么，计算机也就像语文、数学、物理、外语一样，成为现代人所必不可少的基本科学文化知识和必修的课程。这就是所谓“计算机文化”，或者“第二文化”等提法的由来。无论这些提法准确与否，但它所反映的现代社会对现代人知识结构的要求则是无庸置疑的。

1.1.2 用计算机来解决问题

作为理工科的学生，即便是非计算机专业的学生，对于计算机的认识和计算机技术的掌握，当然不能仅仅停留在最粗浅的了解和最基本功能的使用，而必须熟练地用计算机来解决工作中所面临的各种实际问题。

对于组成计算机的硬件系统，包括中心处理器(CPU)、存储器(memory)、输入/输出接口(I/O interface)、外存储器(包括软盘、硬盘、光盘、磁带等)、显示器(monitor)、键盘(keyboard)、鼠标(mouse)、调制/解调器(modem)、声卡(sound blaster)、视卡(vedio blaster)等的电路构成及其工作原理，要有较为透彻的理解。这是《微机原理》等课程所要学习的内容。

计算机的软件大致可以分为系统软件和应用软件两大类。

系统软件是指管理、控制计算机及其外围设备，提供计算机与计算机之间、计算机与用户之间的界面的软件。最常用的系统软件有操作系统、各种计算机语言的编译或解释程序、

计算机网络管理系统等。

应用软件则是为了解决特定的应用问题而制作的软件。

随着计算机应用范围的不断扩大，对科学技术、经济、军事、文化和社会生活的介入程度不断加深，应用软件的开发和生产已发展成为极大的产业。凡是用到计算机的地方，必然要求开发相应的应用软件。

例如，银行、证券、期货、财务等系统必须要有专门的应用软件系统来管理帐目、收支、结帐、核对等。

大型运动会，必然要有应用软件来作比赛日程的安排，运动场馆的管理，运动员食宿、交通、安全等问题的管理，比赛成绩的公布等等。

办公自动化系统中的事务管理、文件管理、文字处理、统计分析、图表制作、决策支持等等。

工业自动化中的自动测量与控制等。

计算机辅助设计(CAD)、辅助制造(CAM)、辅助测量(CAT)、辅助教学(CAI)等等。

语音、图像/图形信号的自动识别、鉴别、压缩、解压等等。

计算机网络通过各种通信手段，将计算机及以计算机为核心的各种系统联系在一起，实现硬件、软件和信息资源的共享，提供电子邮件、信息查询、电子公告、文件下载、电视会议等多种服务。

有关的例子实在是不胜枚举。

在计算机为人们所提供的简捷、迅速、高效、准确、不厌其烦的优质服务的背后，有着大量的应用软件开发人员的辛劳。

一方面是市场广阔、大有所为，另一方面是任务艰巨、竞争激烈。这就要求软件开发人员具有敏捷严密的思维，扎实的软件开发能力，对各种计算机语言及其开发环境的熟悉以至得心应手。

这就是说，同学们作为下一代的科技工作者，必须认真学习和切实掌握一门以上的计算机语言，并熟练地使用来作程序设计，用以解决将要面临的实际问题。

§1.2 程序设计基础

1.2.1 程序设计的步骤和方法

当我们要编制一个程序来解决某一特定的问题时，程序设计的步骤大致可以用图1-1来表示。其主要的步骤为：

1. 分析问题

对要解决的问题做认真仔细的分析，包括弄清楚已知的条件和需要得到的结果及其精度等。有点像求解一个数学应用题时所做的分析。

2. 构造模型

根据对问题的分析，确定要使用的模型。相当于求解应用题时列写的方程式。

3. 确定算法

算法是指解决问题时所执行的一系列步骤。算法选择得恰当与否对能否得到希望的结

果、程序执行的效率、所得结果的精度等影响甚大。鉴于算法在程序设计中的重要地位和作用，我们将在以后的章节中进一步讨论算法的概念和介绍常用算法。

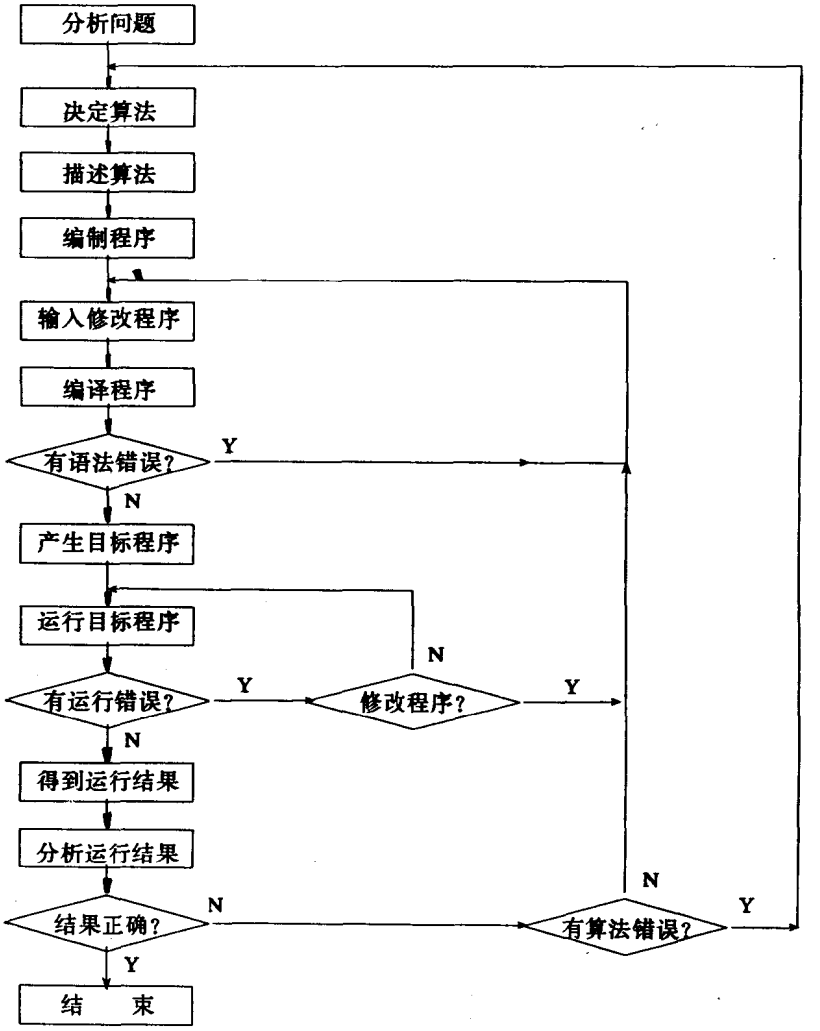


图1-1 程序设计步骤

4. 算法的描述

算法确定以后，程序的编制也就是用计算机语言对算法加以精确地描述。一个较大的程序往往具有较为复杂的结构。为了保证程序编制时的思路清晰，往往用文字或图形的方式将算法中的主要步骤加以描述。最常用的方法有流程图、N-S 图、伪码等。我们将在下面就算法描述的主要方法做进一步的讨论。

5. 确定程序设计语言，编制程序

现在有很多的程序设计语言可供使用，不同的计算机语言各有其特点。程序设计者往往从需要解决的问题出发，选择合适的程序设计语言，根据前面步骤所确定的流程和程序结构，编写出源程序(Source program)。

6. 上机编辑、编译、调试、运行程序

(1)使用编辑程序将源程序输入计算机。

(2)使用编译程序对源程序作编译。如果发生语法错误(Syntax error),则退回编辑系统去修改源程序,然后再进行编译。这样反复进行,一直到没有了语法错误,生成目标程序(Object code)。

(3)运行目标程序。如果发生运行错误(Run-time error),则根据错误的性质做适当的处置。如果是系统设置的问题,例如程序要求打印结果,但打印机并没有连接或没有开机,则对系统进行调整。如果是数据输入的问题,例如需要输入整数的时候输入了实数,则重新运行程序,注意用正确的格式输入。如果是程序本身的问题,则需要退回编辑系统修改源程序,然后再编译之后运行。这样反复进行,一直到没有了运行错误,得到运行结果。

(4)分析运行结果。如果运行结果有问题,则往往需要修改源程序,甚至修改算法,重复以上的步骤,直至得到满意的结果。

1.2.2 算法及其描述方法

算法就是求解问题的一系列步骤。

回顾数学中解应用题的情况。首先是分析、理解题意,弄清楚已知什么,求什么,然后再考虑用什么方法来从已知条件解出结果。这种方法往往落实为列出方程(即所谓的数学模型)。最后,考虑如何求解这个方程,从而得到问题的解。这里,求解该问题的一系列步骤就是算法。如何用最简洁的方法来得到最准确的结果,就是算法研究的目标。

例如,计算1 到100 以内的所有整数的和。当然可以用下式来计算:

$$1+2+3+4+\cdots+99+100$$

但也可以用下式计算:

$$(1+100)\times 50$$

这两种算法都可以得到正确的结果,但两者的效率可大不一样。前者要做99 次加法,后者只须一次加法和一次乘法。当需要求和的整数扩大到1000 时,前者的加法次数也就增加到999 次,而后者仍然只做一次加法和一次乘法。

在这个简单的问题求解中,我们已经可以看到算法对计算效率的影响。除此之外,还要考虑对计算精度所带来的影响。

因此,从开始学习程序设计时,就应十分注意算法设计的学习和实践,养成良好的程序设计的风格和习惯。

一般认为,算法具有以下的五个特性:

1. 有限性

一个算法所包含的操作步骤的个数应该是有限的。无穷多个操作步骤意味着永远不可能达到问题的解决。

2. 确定性

一个算法中的每一个操作步骤都应该是准确定义的,不允许存在二义性,以避免运算或操作的不稳定性。

3. 可行性或有效性

算法中的每一个步骤都应该是可行的,使得算法能够执行,从而得到确定的结果。

4. 输入

一个算法执行时,往往需要初始数据,这些数据可能是从外界输入的,也可能本来就嵌入在算法之中。

5. 输出

算法的目的是求得问题的解决，这就是算法的输出。往往可以把输出理解为“对输入的数据按算法规定的步骤运算之后得到的结果”。

对算法的这些特性的认识和理解，会随着我们对算法的学习和实践而不断地得到加深。

算法设计完成之后，需要按照算法所确定的步骤，编制完成计算机程序。这个过程，实际上就是用计算机语言对算法作准确的描述。在算法比较复杂，程序规模比较大时，直接这样去做往往是比较困难的。人们通过长期的实践，创造了一些方法来描述算法的主要步骤，也就是程序的基本结构。常用的方法有自然语言、图解、伪码等。这些方法，无论是对于构思算法、编制程序，还是阅读、理解、修改、维护已有的程序，都是很好的工具。它们有助于我们很好地把握整个算法的结构和逻辑关系。因此，我们应该努力地学习和掌握这些方法，最终使其成为自己得心应手的工具。

以下简要地介绍几种常用的方法。

1. 自然语言

自然语言也就是人们在长期交往中所形成与使用的日常语言。用自然语言来描述算法容易做到通俗易懂，但往往比较冗长，不容易把复杂的逻辑关系交代清楚，还可能因为描述不准确而出现误解或二义性。

2. 流程图

流程图使用一组特定的几何图形及其连接来表示算法中各步骤的主要功能以及它们之间的 关系。图1-2 是PASCAL 程序的三种基本控制结构的流程图。

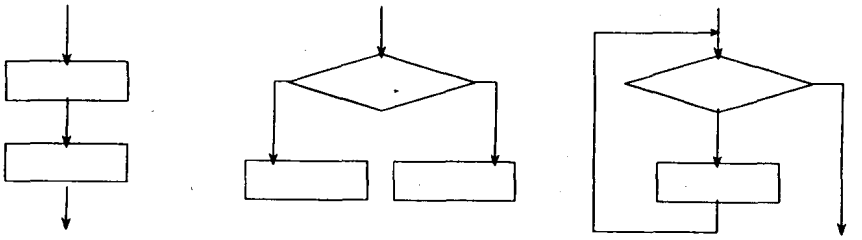
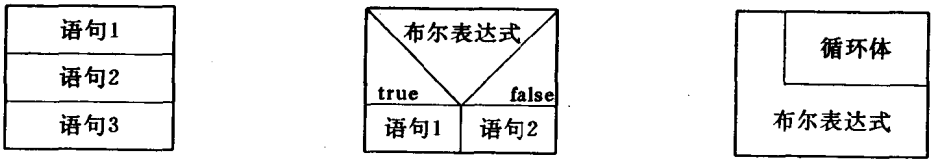


图1-2 三种基本程序结构的流程图

3. N-S 图

使用顺序、选择、循环三种基本结构及其组合，就可以实现任何复杂的算法结构。因此，上述流程图中的连线也可以省去。为此，提出了一种新的流程图——N-S 图。图1-3 是三种基本结构的N-S 图。



顺序结构　　选择结构　　循环结构

图1-3 三种基本结构的N-S 图

4. 伪码

伪码(Pseudocode)也是用语言来 对算法进行描述的。它介乎自然语言和计算机语言之间。伪码没有什么严格的规则,写法比较自由。如果将一个程序看成一篇文章,伪码就像写文章之前所写的提纲,既可以是粗略的要点,也可以相当详细。我们将在以后 的章节中,结合具体的例子来介绍伪码的使用方法。

§1.3 结构化程序设计方法

并不是所有的程序都能得到正确的结果。有多种原因可能造成错误结果:对问题的理解、说明可能不正确;选用的算法不当,不能得到真正的解;程序不是算法的准确体现;程序是正确的,但用了不适当的数据去运行等等。

我们在这里只讨论两种错误:选择了不正确的算法和程序没有准确地实现算法。

上一节讨论算法的基本概念时,已可以清楚地看出,错误的算法必然导致错误的结果。即便是正确的算法,也会给程序的运行效率和结果的精度带来极大的影响。因此,程序的质量主要是由算法决定的。算法的好坏决定了程序的好坏,程序不可能比算法更好。

什么样的算法是好的算法?判断程序好坏的标准又是什么?

在计算机发展的早期,计算机的运算速度慢,存储器容量小。要用它来完成复杂的运算,算法及程序的效率就成为第一重要的事情。为了追求效率的改善,往往使用很多技巧,从而导致程序结构的复杂和混乱,难以阅读、理解、维护,潜伏下错误的因子。

随着计算机科学和技术的飞速发展,速度越来越高,存储器容量也越来越大,对效率的苛求有所缓解。人们也更清楚地认识到,程序的效率,主要是由算法决定的,程序编制的技巧并不能对程序的效率产生决定性的影响。另一方面,程序的规模越来越大,地位也越来越重要。随着程序规模的增大,出错的可能性迅速增大。随着程序地位的越来越重要,程序的错误所带来的后果也就越来越严重,甚至可能是灾难性的。在这样的情况下,判断程序好坏的标准,也就从效率第一,改变成可靠性与可维护性第一了。

为了从根本上保证程序的正确与可靠,就要求软件编制人员改变早期的按个人的习惯和爱好来编制程序的个体手工业方式,代之以软件生产的科学化、规范化、工程化。结构化程序的设计思想和方法由此而产生。

一般认为,结构化程序设计方法包含了以下三方面的内容:

1. 自顶向下(top-down)
2. 逐步细化,或逐步求精(refining)
3. 模块化(model)

一个作者在写一本书时,总是先考虑书的主题和主要内容。然后分章。确定各章的内容。章内分节,写出每节的提纲。然后,按照这个确定好的结构一章一节地写下去。如果说,全书是第一层,章是第二层,节是第三层,节内的各段是第四层,这就是一个自顶向下,逐步细化的过程。

所谓模块化,就是将一个大的程序分成若干个相对独立的小部分(称为模块),每个模块都可以单独设计、调试。模块还可以分为更小的模块。这样,每一个模块的规模较小,完成的任务比较单纯,出错的可能性相对较小,出了错也比较容易查找与修改。另外,程序规模较大

时，往往不是一个人能够独立完成的，需要组织一批人来共同完成。模块化的结构显然有助于程序设计人员的分工合作。在上面讨论的写书的例子中，一本书作为一项总的任务，各章可以看成是相对独立的模块，而章内的各节就是模块中的子模块。在全书的指导思想和写作方法明确之后，相对独立的各章可以分别加以完成。

以上只是对结构化程序设计的意义和方法做了一个最粗浅的介绍。在软件结构与设计方法上，人们通过长期的理论研究和软件工程的实践，已取得了丰硕的成果。其丰富的内容已不是这本教材所能包容的。有志于从事软件工作的读者可以继续学习有关的课程。

§1.4 PASCAL 语言

1.4.1 计算机语言

语言是人们用以交流思想与信息的工具。只有通过双方都能理解的语言，或者通过对双方的语言都能理解的翻译，交流才能成功。当人们使用计算机，命令它进行预定的操作时，首先必须使计算机“理解”用户所发出的命令，然后才有可能去执行。由此而产生计算机语言的问题。

1. 机器语言(Machine Language)

众所周知，数字计算机中的各种元器件只有“通”、“断”两种状态，分别用“0”、“1”来表示。因此，计算机只能接受并执行用二进制方式编写的指令，称为机器指令。每种计算机机器指令的序列就称为这种计算机的机器语言程序。机器语言属于我们通常所说的低级语言(Low-level language)

下面是一段用机器语言编写的程序：

```
00111010
01100000
00000000
00111100
00000000
00110010
01100000
00000000
```

它命令计算机把地址为01100000的存储单元中的数调入运算器，将其加1，把结果再存回原来的存储单元。

这段程序所规定的操作非常简单。但是，这种二进制代码的形式是很难为人们，尤其是非计算机专家所能理解和记忆的，因为它离我们日常使用的自然语言相距甚远。此外，每种计算机的机器语言都不相同。也就是说，为一种计算机编写的程序，在另外一种计算机上是无法运行的。

2. 符号语言(Symbolic language)

鉴于机器语言的难以理解和记忆，人们使用一些符号来代表相应的二进制代码，这就是符号语言，也称为汇编语言。例如，上面给出的从某存储单元取出一个数，加1后再存回去的